
IF-60/70/80 用
Windows API・USB デバイスドライバ
Ver. 3.10

＜ソフトウェア使用許諾契約＞

1. 権利の許諾

当社はお客様に対して、本使用許諾契約に同意いただいてダウンロード可能となるソフトウェア及びその関連資料(以下「本ソフトウェア」といいます)に関し、以下の権利を許諾します。

- (a) お客様は、本ソフトウェアに対応する当社製品を利用する目的で本ソフトウェアを使用することができます。
- (b) お客様は、本ソフトウェアを複製し、1台以上のコンピュータ上で使用することができます。

2. 追加許諾条項

本ソフトウェアを定められた目的に従って使用した結果、作成された各種のファイルは、お客様の著作物となります。

3. 著作権

本ソフトウェア及びその複製物の著作権は当社又は当社が認めた者が有するものであり、日本国著作権法及び国際条約によって保護されています。本使用許諾契約に基づき、お客様が本ソフトウェアを複製する場合は、ダウンロードされた本ソフトウェアに付されていたものと同じの著作権表示がなされることを要します。

4. 禁止事項

本ソフトウェアがソースコードで提供される場合、お客様は、ソースコードを改変したものを当社製として第三者に配布することはできません。

5. 無保証

当社は、本ソフトウェアがお客様特定の目的のために適切であること、もしくは有用であること、又は本ソフトウェアに瑕疵がないこと、その他本ソフトウェアに関していかなる保証もいたしません。

6. 免責

当社は、いかなる場合においても、本ソフトウェアの使用又は使用不能から生ずるいかなる損害(事業利益の損害、事業の中断、事業情報の損失、又はその他金銭的損害)に関して、一切責任を負いません。

7. 契約の解除

お客様が本使用許諾契約に違反した場合、当社は本使用許諾契約を解除することができます。その場合、お客様は本ソフトウェアを一切使用しないものとします。

＜サポート＞

本ソフトウェアの不具合等のお問い合わせは、当社サービスまでお願いいたします。

＜保証範囲＞

本ソフトウェアのダウンロード・インストールはお客様の責任においておこなっていただきます。また本ソフトウェアは、予告せず改良、変更することがあります。

＜著作権者＞

各ソフトウェアの著作権は、当社に帰属します。

＜各社商標＞

TEXIOは当社の産業用電子機器における製品ブランドです。また、本説明書に記載されている会社名および商品名は、それぞれの国と地域における各社および各団体の商標または登録商標です。

目 次

ソフトウェア使用許諾契約

第 1 章 概 要	1
第 2 章 特 徴	1
第 3 章 動作環境	1
第 4 章 API のインストール	1
第 5 章 USB デバイスドライバのインストール	1
5-1. インストールについて	1
5-2. 再インストールについて	2
第 6 章 関数一覧表	2
6-1. VC++ 利用時の関数一覧	2
6-2. VC# 利用時の関数一覧	2
6-3. VB6.0 及び VBA 利用時の関数一覧	3
第 7 章 デバイス制御関数	4
7-1. USB488_DeviceOpen	4
7-2. USB488_DeviceClose	4
7-3. USB488_SetTimeOut	4
7-4. USB488_SetRemote	4
7-5. USB488_GetRemote	5
7-6. USB488_DevClear	5
7-7. USB488_GetSTBQuery	5
7-8. USB488_ClearSTB	5
7-9. USB488_Send	6
7-10. USB488_SendBinary	6
7-11. USB488_Receive	6
第 8 章 利用上の注意について	7

第 1 章 概要

- ・本ソフトウェアは、USB 用デバイスドライバとアクセス用のモジュール (API) で構成され、IF-60/70/80 シリーズを装着した電源および電子負荷を WindowsPC から USB でコントロールすることが可能です。

第 2 章 特徴

- ・各種言語用のヘッダファイルとサンプルを用意してあるため、容易にプログラム開発が可能です。

第 3 章 動作環境

- ・マイクロソフト社の Windows7 以降(32bit / 64bit)が動作する PC。
- ・USB デバイスドライバは、弊社が作成したものを使用します。PC 側の USB 動作環境は USB1.1 および USB2.0 のフルスピードになります。
- ・対応言語として、Visual Basic6.0、Visual Basic for Applications、Visual C++6.0(C,C++)、VisualBasic.2008、VisualC++.2008、VisualC#.2008 以後を想定しています。
- ・サンプルプログラムで GP-IB を使用する場合は National Instruments 社製 NI488.2M SoftWare が動作する GP-IB カード。(NI488.2M Software は、Ver.2.8.1 以上を使用してください。)

第 4 章 API のインストール

- ・PC へのログインは必ず管理者(administrator)としてください。
- ・適当なフォルダにダウンロードしたファイルを展開し、API フォルダにある 32 ビット・64 ビットの環境に合わせた Setup32.exe(32bit 用)／Setup64.exe(64bit 用)を右クリックで管理者として実行しインストールしてください。
- ・USB を使用する場合は続けて Driver フォルダにある Setup*.exe を実行し、デバイスドライバをインストールしてください。
- ・VisualBasic6.0,VBA(Excel2000 以上)を使用するときは、IF_*0.bas を標準モジュールに追加登録することで API を使用できます。VisualBasic2005 以後を使用するときは、IF_*0.vb を追加登録することで API を使用できます。
- ・Visual C++、C#についてはサンプルプログラムを参照してください。。
- ・API の利用方法は、サンプルプログラムを用意してありますのでこちらを参照してください。
- ・OS によってはインストール時に設定または続行の確認メッセージが出ますが、そのまま続行してください。
- ・インストール中にフォルダの指定が出る場合がありますが、インストール用のフォルダは作成されません、ファイルは全てシステムフォルダにコピーされます。
- ・.netFramework4 ライブラリと VC++ランタイムライブラリのインストールが自動で行われます。システムと競合してインストールできない場合はライブラリを一旦削除してインストールするか TEXIO_API32.msi または、TEXIO_API64.msi を実行して下さい。
- ・再インストールは環境に応じて TEXIO_API32.msi または、TEXIO_API64.msi を実行して下さい。

第 5 章 USB デバイスドライバのインストール

5-1.インストールについて

- ・本体の電源スイッチを OFF にしたままで USB ケーブルを動作している PC に接続してください。
- ・PC へのログインは必ず管理者(administrator)としてください。
- ・Driver フォルダにある 32 ビット・64 ビットの環境に合わせた Setup32.exe(32bit 用)／Setup64.exe(64bit 用)を右クリックで管理者として実行しインストールしてください。途中で マイクロソフトのロゴテストに合格していない とメッセージが出ることがありますが、そのまま続行してください。
- ・インストールが完了してから、本体の電源スイッチを ON して PC が認識するのを待ちます。接続している機器の台数や状況により時間がかかる場合があるので注意してください。
- ・認識された場合、新しいハードウェアの検出が始まります。自動でインストールを選択し、メッセージにしたがってデバイスドライバをインストールします。途中で マイクロソフトのロゴテストに合格していない とのメッセージが出ることがありますが、そのまま続行してください。OS によってはインストール時に設定または続行の確認メッセージが出ますが、そのまま続行してください。
- ・ドライバが正常にインストールできなかった場合はデバイスマネージャーから再度インストールをおこなってください。
- ・PC によってはセキュリティの関係で本器が自動認識されないことがあります。この場合はコントロールパネルのデバイスマネージャーで"ほかのデバイス"にある本器を右クリックで選択し、ドライバの更新を行ないます。ドライバファイルは手動で同梱の inf ファイルを指定してください。

5-2.再インストールについて

- ・本ドライバを使用している全てのデバイスとの通信が停止している事を確認してください。
- ・「コントロールパネル→プログラムの追加と削除」からドライバを削除してください。
(Windowsドライバパッケージ (WinUSB)USB Driver の当社発行のもの)
- ・ドライバを Setup**.exe で再度インストールしてください。
- ・デバイスマネージャで、接続している機器が正常に認識・動作していることを確認してください。

第 6 章 関数一覧表

6-1. VC++利用時の関数一覧

	関 数	内 容
1	Long USB488_DeviceOpen(int Addr)	デバイスのオープン
2	int USB488_DeviceClose(int Addr)	デバイスのクローズ
3	int USB488_SetTimeout(int Addr, int tTime)	タイムアウトの設定
4	int USB488_SetRemote(int Addr, int iMode)	リモート・ローカルの設定
5	int USB488_GetRemote(int Addr)	リモート・ローカルの読出
6	int USB488_DevClear(int Addr)	デバイスのクリア
7	int USB488_GetSTBQuery(int Addr)	ステータスバイトの読出
8	int USB488_ClearSTB(int Addr)	ステータスバイトのクリア
9	int USB488_Send(int Addr, char * sMsg)	文字列の送信
10	int USB488_SendBinary(int Addr, char * sStream, int iCount, int iMode)	バイナリデータの送信
11	int USB488_Receive(int Addr,int Count, char *sReceive)	データの受信

VC++より API を利用する場合は、DLL インポート用の定義ファイルを参考にしてください。

サンプルプログラムでは、関数名は“USB488_”をはずした名前に登録してあります。

利用時にイニシャライズを実行し DLL の実関数を割当ててください。

6-2. VC#利用時の関数一覧

	関 数	内 容
1	Int USB488_DeviceOpen(int Addr)	デバイスのオープン
2	int USB488_DeviceClose(int Addr)	デバイスのクローズ
3	int USB488_SetTimeout(int Addr, int tTime)	タイムアウトの設定
4	int USB488_SetRemote(int Addr, int iMode)	リモート・ローカルの設定
5	int USB488_GetRemote(int Addr)	リモート・ローカルの読出
6	int USB488_DevClear(int Addr)	デバイスのクリア
7	int USB488_GetSTBQuery(int Addr)	ステータスバイトの読出
8	int USB488_ClearSTB(int Addr)	ステータスバイトのクリア
9	int USB488_Send(int Addr, string strBuf)	文字列の送信
10	int USB488_SendBinary(int addr, [In] byte[] pmsg,int count, int nullMode)	バイナリデータの送信
11	int USB488_Receive(int addr, int cnt, StringBuilder pmsg) int USB488_Receive(int addr, int cnt, [In, Out] byte[] bytes)	データの受信

VC#より API を利用する場合は、DLL インポート用の定義ファイルを参考にしてください。

6-3. VB6.0 及び VBA 利用時の関数一覧

	関 数	内 容
1	USB488_DeviceOpen (ByVal Addr As Long) As Long	デバイスのオープン
2	USB488_DeviceClose (ByVal Addr As Long) As Long	デバイスのクローズ
3	USB488_SetTimeout (ByVal Addr As Long, ByVal sec As Long) As Long	タイムアウトの設定
4	USB488_SetRemote (ByVal Addr As Long, ByVal rMode As Long) As Long	リモート・ローカルの設定
5	USB488_GetRemote (ByVal Addr As Long) As Long	リモート・ローカルの読出
6	USB488_DevClear (ByVal Addr As Long) As Long	デバイスのクリア
7	USB488_GetSTBQuery (ByVal Addr As Long) As Long	ステータスバイトの読出
8	USB488_ClearSTB (ByVal Addr As Long) As Long	ステータスバイトのクリア
9	USB488_Send (ByVal Addr As Long, ByVal strbuf As String) As Long	文字列の送信
10	USB488_SendBinary (ByVal Addr As Long, strbuf As Any, ByVal Count As Long, iMode As Long) As Long	バイナリデータの送信
11	USB488_Receive (ByVal Addr As Long, ByVal Count As Long, ByVal strbuf As String) As Long	データの受信
12	USB488_ReceiveB (ByVal Addr As Long, ByVal Count As Long, strbuf As Any) As Long	バイナリデータの受信

VisualBasic6.0 または、VBA より API 関数を利用する場合は、source フォルダの if_*.bas ファイルを利用してください。

VisualBasic2005 以後の場合は、if_*.vb を利用してください。

VisualBasic2005 以後での関数定義は、言語仕様の変更に伴い

Long 宣言 → Integer 宣言

Any 宣言 → Byte() 宣言

に変更されています。

第 7 章 デバイス制御関数

7-1. USB488_DeviceOpen

機能 : 指定されたデバイスについて、オープン処理を行ないます。

C 書式 : long USB488_DeviceOpen(int Addr)

VB 書式 : USB488_DeviceOpen (ByVal Addr As Long) As Long

パラメータ:

- ・ ADDR: 通信を開始する機器の PC アドレスを指定します。

戻り値 :

正常にデバイスをオープンできた場合は、Windows のデバイスハンドルとなります。

デバイスが見つからない場合は、-1 となります。

注意事項 :

特に無し。

7-2. USB488_DeviceClose

機能 : 指定されたデバイスについて、クローズ処理を行ないます。

C 書式 : int USB488_DeviceClose(int Addr)

VB 書式 : USB488_DeviceClose (ByVal Addr As Long) As Long

パラメータ:

- ・ Addr: 通信を終了する機器の PC アドレスを指定します。

戻り値 :

関数が成功した場合、0 が戻り、エラーが発生した場合は、-1 が戻ります。

注意事項 :

プログラム終了時は、必ずデバイスをクローズしてください。また、連続で高速に OPEN と CLOSE を繰り返すと通信がデッドロックする場合がありますので注意が必要です。

7-3. USB488_SetTimeOut

機能 : 指定されたデバイスについて、通信タイムアウト設定を行ないます。

C 書式 : int USB488_SetTimeOut(int Addr, int tTime)

VB 書式 : USB488_SetTimeout (ByVal Addr As Long, ByVal sec As Long) As Long

パラメータ:

- ・ Addr: 設定する機器の PC アドレスを指定します。
- ・ Time: タイムアウト時間指定(単位—ミリ秒)。0~65535。
0 秒指定で、タイムアウト検出を禁止します。初期値は、1000ms になります

戻り値 :

関数が成功した場合、0 が戻り、エラーが発生した場合は、-1 が戻ります。

注意事項 :

ローカルバスを利用する場合は、タイムアウトを長めに設定する必要があります。

7-4. USB488_SetRemote

機能 : 指定されたデバイスについて、リモート・ローカル設定を行ないます。

C 書式 : int USB488_SetRemote(int Addr, int iMode)

VB 書式 : USB488_SetRemote (ByVal Addr As Long, ByVal rMode As Long) As Long

パラメータ:

- ・ Addr: 設定する機器の PC アドレスを指定します。
- ・ iMode: ローカル・リモート・ローカルロックアウトの状態を切替えます。
0: ローカル状態
1: リモート状態 (ローカルキー有効)
2: リモート状態 (全キー無効)

戻り値 :

関数が成功した場合、0 が戻り、エラーが発生した場合は、-1 が戻ります。

注意事項 :

- ・ 特に無し。

7-5. USB488_GetRemote

機能 : 指定されたデバイスについて、リモート・ローカル状態を読み出します。

C 書式 : int USB488_GetRemote(int Addr)

VB 書式 : USB488_GetRemote (ByVal Addr As Long) As Long

パラメータ:

- ・ Addr: 設定する機器の PC アドレスを指定します。

戻り値 :

関数が成功した場合、正値が戻り、エラーが発生した場合は、-1 が戻ります。

-1: デバイスエラー

0: ローカル状態

1: リモート状態(ローカルキー有効)

2: リモート状態(全キー無効)

注意事項 :

- ・ 特に無し。

7-6. USB488_DevClear

機能 : 指定されたデバイスにデバイスクリアを発行します。

C 書式 : int USB488_DevClear(int Addr)

VB 書式 : USB488_DevClear (ByVal Addr As Long) As Long

パラメータ:

- ・ Addr: 設定する機器の PC アドレスを指定します。

戻り値 :

関数が成功した場合、0 が戻り、エラーが発生した場合は、-1 が戻ります。

注意事項 :

デバイス側は、通信バッファをクリアします。また、ローカルバスを利用している場合は、ローカルバスのアドレス指定は解除されます

7-7. USB488_GetSTBQuery

機能 : 指定されたデバイスについて、ステータスバイトを読み出します。

C 書式 : int USB488_GetSTBQuery(int Addr)

VB 書式 : USB488_GetSTBQuery (ByVal Addr As Long) As Long

パラメータ:

- ・ Addr: 設定する機器の PC アドレスを指定します。

戻り値 :

関数が成功した場合、ステータスバイトが戻り、エラーが発生した場合は、-1 が戻ります。

ステータスバイトの内容は、本体取扱説明書を参照してください。

注意事項 :

- ・ 特に無し。

7-8. USB488_ClearSTB

機能 : 指定されたデバイスについて、ステータスバイトをクリアする。

C 書式 : int USB488_ClearSTB(int Addr)

VB 書式 : USB488_ClearSTB (ByVal Addr As Long) As Long

パラメータ:

- ・ Addr: 設定する機器の PC アドレスを指定します。

戻り値 :

関数が成功した場合、0 が戻り、エラーが発生した場合は、-1 が戻ります。

注意事項 :

- ・ 特に無し。

7-9. USB488_Send

機能 : 指定されたデバイスについて、文字列を送信します。

C 書式 : int USB488_Send(int Addr, char * sMsg)

VB 書式 : USB488_Send (ByVal Addr As Long, ByVal strbuf As String) As Long

パラメータ:

- ・ Addr: 設定する機器の PC アドレスを指定します。
- ・ Smsg: 送信する文字列を指定します。

戻り値 :

関数が成功した場合、転送バイト数が戻り、エラーが発生した場合は-1 が戻ります。

注意事項 :

文字列は、必ず NULL で終端していることが必要です、通常は LF などのデリミタは不要です。

配列などで文字列を用意する場合は、必ず 0 を最後に入れ、転送バイト数を確認してください。また、使用言語によって、文字列は ASCII でなく UNICODE の場合があります。この場合は、文字コードの変換が必要になります。

7-10. USB488_SendBinary

機能 : 指定されたデバイスについて、文字列を送信します。

C 書式 : int USB488_SendBinary(int Addr, char * sStream, int iCount, int iMode)

VB 書式 : USB488_SendBinary (ByVal Addr As Long, strbuf As Any, ByVal Count As Long, iMode As Long) As Long

パラメータ:

- ・ Addr : 設定する機器の PC アドレスを指定します。
- ・ sStream : 送信するバッファを指定します。
- ・ iCount : 送信するバイト数を指定します。
- ・ iMode : 送信終了のヌルパケット付加のフラグ。1:付加あり

戻り値 :

関数が成功した場合、転送バイト数が戻り、エラーが発生した場合は、-1 または-2 が戻ります。

注意事項 :

通信の完了までは制御が戻りません。大量データで時間がかかる場合は、分割して転送するか別のスレッドで転送し、途中で OS に制御を戻すようにしてください。

7-11. USB488_Receive

機能 : 指定されたデバイスについて、データを受信します。

C 書式 : int USB488_Receive(int Addr, int Count, char *sReceive)

VB 書式 : USB488_Receive (ByVal Addr As Long, ByVal Count As Long, ByVal strbuf As String) As Long

: USB488_ReceiveB (ByVal Addr As Long, ByVal Count As Long, strbuf As Any) As Long

パラメータ:

- ・ Addr : 設定する機器の PC アドレスを指定します。
- ・ Count : 受信要求のバイト数を指定します。
- ・ sReceive : 受信するバッファを指定します。

VB からは受信バッファの形式により USB488_Receive と USB488_ReceiveB の関数が用意されています。

戻り値 :

関数が成功した場合、転送バイト数が戻り、エラーが発生した場合は、-1 または-2 が戻ります。

注意事項 :

受信バッファは関数を呼ぶ前にサイズを確定し、オーバーフローしないように設定してください。

受信バッファの未転送部分のデータは不定となっています。必ず先頭から転送バイト数分切り出して利用してください。

第 8 章 利用上の注意について

- PC の CPU クロックが高速な場合に、設定や読み取りを連続で行うと、本体の動作速度が遅くなる場合があります。
また、同じ設定のオン・オフを繰り返して行くと、処理が終わる前に次の設定が行われるため、設定が省略されて見えることがあります。特に内部のステータスやモニタデータの読出しについては、数百 ms の読取周期がありますので注意が必要です。
- 本 API は、サスペンドなどの PC の省電力モードに対応していません。省電力モード以後の本体の動作は、不定となりますので設定には十分注意してください。
- 通信を行う場合に、PC および機器のグランド電位が異なったり、接地されていない場合は、動作が不安定になります。必ず 3 芯の電源ケーブルを利用し、筐体を接地した状態でご利用ください。USB ハブを利用する場合も、できるだけ筐体グランドが同じ電位になるように設置してください。
- モーターやインバーター、溶接機などノイズが発生している環境では、通信が不安定になります。PC、被制御機器、通信経路、商用電源がノイズの影響を受けないように設置し、十分に動作確認をした上でご利用ください。
- ご利用になる言語によっては 2 進数演算による浮動小数点データが正しく表示されない場合があります。数値の比較を行う場合は、ご利用の言語のデータ形式をよく確認のうえ、正しい値が得られるようにプログラムしてください。
- 電源を複数のプログラム、プロセス、スレッドから同時にコントロールする場合は通信が重ならないように十分検証をおこなった上でご使用ください。
- ローカルバス接続を行なう機種の場合、スレーブが複数存在すると、通信が遅くなる場合があります。送信の間隔を広げて通信が重ならないようにご使用ください。
- アプリケーションのフォルダなどに、API の "IF_*0.dll" がある場合は、動作がおかしくなる場合があります。フォルダから DLL を削除して API を正しくインストールしてください。
- WindowsUpdate によるインストールが実行中の場合は API または USB デバイスドライバのインストールができないことがあります、WindowsUpdate の処理がすべて終了してから再度インストールをおこなってください。
- OS や USB の仕様変更に伴い接続断やノイズなどの状態異常時の各関数の戻り値や動作が正しくない場合があります。



株式会社 テクシオ・テクノロジー

〒222-0033 神奈川県横浜市港北区新横浜 2-18-13 藤和不動産新横浜ビル 7F
<https://www.texio.co.jp/>

アフターサービスに関しては下記サービスセンターへ

サービスセンター 〒222-0033 神奈川県横浜市港北区新横浜 2-18-13 藤和不動産新横浜ビル 8F
TEL.045-620-2786